An Exploration of Year Prediction Using Large Scale Data

E. Indhuma
Assistant Professor, Dept of Computer Science and Engineering
St.Annes College of Engineering and Technology

Abstract- *Music, melodies, and recordings are plentiful on the web and continue to develop. The enormous scope music documents rely upon computerized calculations to dissect and file music content. Thus consequently extricating music data is acquiring significance as an approach to put together and structure music records. Music data recovery (MIR) is an arising research region to adapt to such need. The MIR undertakings incorporate feeling acknowledgment, cover melody acknowledgment, sort characterization, craftsman ID, instrument acknowledgment, and music comments. Since present day music industry is developing quickly there is need for quicker music examination and order frameworks which are fit for taking care of enormous scope informational collections with million tunes. Highlight extraction and classifier learning are the vital parts of music data recovery. There are two fundamental difficulties in applying existing strategies to huge scope information. Versatility is the principal significant issue. Extraction of sound highlights for enormous measure of information is a tranquil tedious interaction. The subsequent trouble is making progress truth data for enormous scope information. This examination work centers around creating productive calculations and strategies for huge scope MIR errands with Million Song Dataset (MSD). MSD is a uninhibitedly accessible assortment of sound highlights and metadata for 1,000,000 contemporary mainstream melodic tracks. Mel Frequency Cepstral Coefficients (MFCC) and straight relapse are utilized to foresee the delivery year of the tune in a given arrangement of highlights.*

Keywords- Million Song Dataset, Machine learning, Mel-frequency Cepstral Coefficient, Linear Regression, Year Prediction.

I. INTRODUCTION

Current music arrangement frameworks are generally tried on little or middle size informational indexes with a large number of melodies. Current music industry is developing quickly. To keep track with the always expanding number of new tunes made accessible regularly by the two experts and mysterious creators, quicker music investigation and characterization frameworks equipped for taking care of huge scope informational collections with a great many melodies are required. This postures two fundamental difficulties to the current methods for music characterization. Versatility is the main significant issue in regards to both handling time and capacity. Highlight extraction for music sound examination is a very tedious cycle. The extraction of standard sound highlights, as MFCC, pitch, and cadenced highlights, requires applying ghastly change like FFT to neighborhood outlines as the fundamental advance. This is very exorbitant in any event, for tunes with moderate lengths. To speed up enormous scope music arrangement assignments, quicker pre-handling strategies are expected to smooth out include extraction. Million Song Dataset (MSD), a wholeheartedly available assortment of sound features and metadata for 1,000,000 present day notable melodic tracks, is used in this work. MSD helps specialists by giving a huge scope dataset. The managed learning pipeline depicted in past part is carried out utilizing the Python programming interface to Spark (pySpark). PySpark gives a simple to-utilize programming reflection and equal runtime. Tough Distributed Datasets (RDDs) are the key idea. Relapse is a bunch of methods for assessing connections. One of the least difficult kind of relationship is direct which is called as straight relapse, and it has numerous applications. Regardless of its straightforwardness, direct relapse is an amazingly integral asset for examining information. Execution of the Root Mean Square Error for both the benchmark model and straight relapse model. Million melody dataset is utilized for the trials and the framework is created utilizing Python programming interface to Spark (pySpark). The LR model has benchmark and Regression model.

II. PROPOSED SYSTEM

A. YEAR PREDICTION SYSTEM

A supervised learning based framework is proposed to anticipate the delivery year of the melody. Managed taking in takes in a planning from substances to nonstop marks given a preparation set. In this work sound highlights are planned to year of the tune as demonstrated in Figure 2.1. The crude information is parted into training set and test set. The preparation set is to prepare the model and the test set is to assess the precision of the prepared model. The sound

highlights are separated from the preparation set and used to prepare different models. Test set assesses last model's precision. Last model would then be able to be utilized to mention forecasts on future objective facts, new tunes in this work.

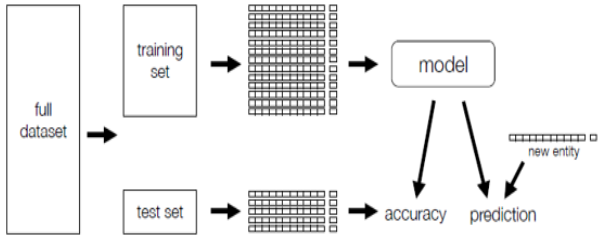

Figure: 2.1 Supervised learning pipeline for year prediction

B. MSD DATASETS

Million Song Dataset (MSD), may be a freely-available collection of sound highlights and metadata for a million modern well known musictracks. MSD makes a difference analyst by giving a large-scale dataset. The MSD contains

- 280 GB of information
- 1,000,000 songs/files
- 44,745 special artists
- 7,643 one of a kind terms (Echo Nesttags)
- 2,321 special music brainz labels
- 43,943 specialists with at slightest one term
- 2,201,916 deviated closeness connections
- 515,576 dated tracks beginning from1922
- 18,196 cover song recognized.

The data is stored utilizing HDF5 organize to proficiently handle the heterogeneous sorts of information such as sound highlights in variable cluster lengths , names as strings, longitude/latitude, comparable artists, etc. . Each tune is depicted by a single record, a list of all areas accessible within the records of the dataset. Parsing 1,000,000 files to recognize some tens of thousands different artists is simply an awesome way to show off various machine learning skills.

C. FEATURE EXTRACTION

Feature extraction might be a typical term for strategies for building mixes of the elements to get around these issues though as yet depicting the data with sufficient exactness. It tends to the issue of how to address the outlines to be ordered as far as incorporate vectors or pairwise comparable qualities.

The removed highlights should have explicit qualities are

i. easily quantifiable, happen normally and habitually in discourse,
ii. not change over the long haul, fluctuate as much among speakers,
iii. consistent for every speaker, not influenced by: speaker wellbeing, foundation commotion,
iv. many calculations to extricate them LPC,LPCC,HFCC,MFCC.

Mel-Frequency cepstral coefficients (MFCC) are utilized to address sound records in this work. Primary explanation of the MFCC processor is to copy the conduct of the human ears. MFF Human hearing isn't similarly delicate at all recurrence groups less touchy at higher frequencies generally over 1000 Hz.Mel scale is direct under 1000Hz.

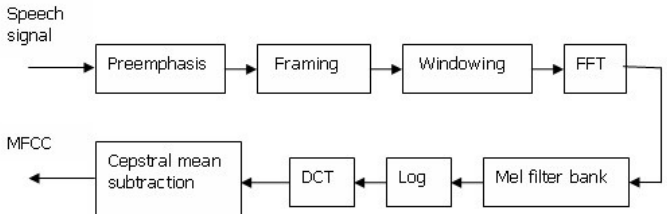

Figure. 2.2. Extraction of MFCC features

Furthermore, logarithmic over 1000 Hz. Tested signs can catch all frequencies up to 5 kHz, which cover most essentialness of sounds that are created by people.

MFCC have been broadly utilized in applications like discourse acknowledgment. It is a portrayal of momentary force range of a sound. It depends on a direct cosine change of a log power range on a nonlinear mel size of recurrence. Each transient Fourier change (STFT) greatness coefficients expanded by the contrasting channel pickup and the comes about are collected. The channels used are three-sided and they are likewise scattered along the Mel-scale, which is portrayed by $Mel(f) = 2595\ log_{10}\left(1 + \frac{f}{700}\right)$

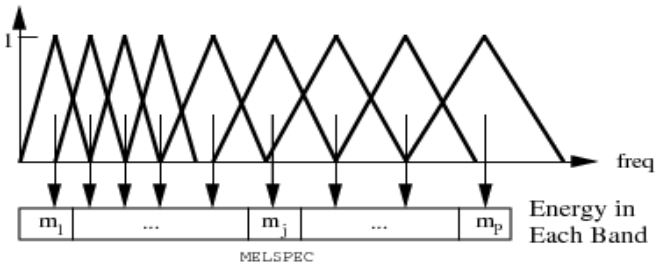

Fig. 2.3. Mel-scale triangular filters

At that point discrete cosine Transform (DCT) is associated with the log of the mel spectral coefficients to get Mel Frequency cepstral coefficients (MFCC). Estimation of the MFCCs inlcudes the accompanying Steps:

- Preemphasis separating
- Take the supreme worth of the STFT (use of Hamming window)
- Warp to hear-able recurrence scale (Mel/Bark)
- Take the DCT of the log-hear-able range
- Return the first ncep parts

After include extraction, changes are prepared to improve execution. Highlight changes upgrades the discriminative abilities of removed highlights. It applies dimensionality decrease techniques and lessens the parts of highlights. The sound highlights are removed from the preparation set and used to prepare various models.

III. MODELLING AND ANALYSIS

Subsequent to procuring information with numerous factors, one vital inquiry is the manner by which the factors are connected. Relapse is a bunch of strategies for assessing connections. One of the easiest sort of relationship is straight which is called as direct relapse, and it has numerous applications. Regardless of its straightforwardness, direct relapse is a unimaginably integral asset for examining information. Straightforward direct relapse is utilized when there are just two factors of interest (e.g., weight and tallness, or power utilized and distance extended). When there are more factors different direct relapse is utilized. We will fit a line y= β_0+β_1 x to our information. Here,x is known as the free factor or indicator variable, and y is known as the reliant variable or reaction variable.β_1 is the incline of the line: this is perhaps the main amounts in any direct relapse investigation. A worth extremely near 0 demonstrates next to zero relationship; huge positive or negative qualities show enormous positive or negative connections, separately. β_ 0is the catch of the line. A generally utilized strategy in measurements to really fit a line is proposing a probabilistic model and utilizing the likelihood of information to assess how great a specific model is A probabilistic model for straightly related data. Let the combined information focuses are $(x_1, y_1),\ (x_2, y_2), \cdots, (x_n, y_n)$, where it is assumed that as a function of $x_i$each $y_i$is generated by using some true underlying line $y = \beta_0 + \beta_1 x$ and then adding some Gaussian noise. Formally$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$. Here, the noise $\varepsilon_i$represents the fact that the data won't fit the model perfectly.ε_i is modeled as being Gaussian: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Note that the intercept β_0 the slopeβ_1, and the noise varianceσ^2 are all treated as fixed (i.e., deterministic) but unknown quantities.

least-squares regression, Assuming that this is actually how the data $(x_1, y_1), \ldots, (x_n, y_n)$ are generated, then it turns out that how to find the line for which the probability of the data is highest by solving the following optimization problem:

$$\min_{\beta_0, \beta_1} \sum_{i=1}^{n} [y_i - (\beta_0 + \beta_1 x_i)]^2,$$

Where $\min \beta_0, \beta_1$means "minimize over $\beta_0$and β_1".This is known as the least-squares linear regression problem. Given a set of points, the solution is:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)2}$$

$$= r\frac{s_y}{s_x}$$

$$\beta_0 = \bar{y} - \beta_0 \bar{x},$$

Where $\bar{x}, \bar{y}, s_x \text{ and } s_y$and are the sample means and standard deviations for *x* values and *y*values, respectively, and *r* is the correlation coefficient, defined as

$$r = \frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{x_i - \bar{x}}{s_x}\right)\left(\frac{y_i - \bar{y}}{s_y}\right)$$

By examining the second equation for the estimated slope $\hat{\beta}_1$, it is observed that since sample standard deviationss_x ands_y are positive quantities, the correlation coefficient *r*,which is always between -1 and 1, measures how much *x* is related to *y* and whether the trend is positive or negative. The square of the correlation coefficient r^2 will always be positive and is called the coefficient of determination. Multiple Linear Regression

Consider the case when instead of just a single scalar value *x*, there is a vector $(x_1, \cdots, x_p)$for every data point *i*. So, there are *n* data points (just like before), each with *p* different predictor variables or features. It is necessary to predict *y* for each data point as a linear function of the different *x* variables:

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p,$$

Even though it's still linear, this representation is very versatile. Let the input data is in matrix form as X, an *x* x *p*

matrix where each row corresponds to a data point and each column corresponds to a feature. Since each output y_i is just a single number, the collection is represented as an *n*-element column vector *y*. Then our linear model can be expressed as

$$y = X\beta + \varepsilon$$

Where β is a *p*-element vector of coefficients, and ε is an *n*-element matrix where each element, like ε_i earlier, is normal with mean 0 and variance σ^2. Notice that in this version, a constant term like β_0 from before is not explicitly included. Instead a column of 1s is added to the matrix X to accomplish this. This leads to the following optimization problem: $\min_\beta \sum_{i=1}^{n}(y_i - X_i\beta)^2$

Where $\min_\beta$ just means "find values β of that minimize the following", and X_i refers to row *i* of the matrix X. Some basic linear algebra can be used to solve this problem and find the optimal estimates:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

From this, confidence intervals and/or hypothesis tests for each coefficient can be obtained. It's important not to blindly test whether all the coefficients are greater than zero: since this involves doing multiple comparisons, appropriate correction is needed using Bonferroni correction or FDR correction. But before even doing that, it's often smarter to measure whether the model even explains a significant amount of the variability in the data: if it doesn't, then it isn't even worth testing any of the coefficients individually.

Typically, an analysis of variance (ANOVA) test is used to measure this. If the ANOVA test determines that the model explains a significant portion of the variability in the data, then there is a need for testingeach of the hypotheses and correcting for multiple comparisons. It is also possible to ask about which features have the most effect: if a feature's coefficient is 0 or close to 0, then that feature has little to no impact on the final result.

There is a need to avoid the effect of scale: for example, if one feature is measured in feet and another in inches, even if they're the same, the coefficient for the feet feature will be twelve times larger. In order to avoid this problem, the standardized coefficients $\frac{\hat{\beta}_k}{s_{\hat{\beta}_k}}$ are used.

A. MODEL EVALUATION

Suppose for a moment that every point y_i was very close to the mean $\bar{y}$: this would mean that each y_i wouldn't depend on x_i, and that there wasn't much random error in the value either. Since this shouldn't be the case, it is necessary to understand how much the prediction from x_i and random error contribute to y_i. In particular, to know how far y_i is from the mean $\bar{y}$. This difference can be written as

$$y_i - \bar{y} = (\hat{y}_i - \bar{y}) + (y_i - \hat{y}_i)$$

In particular, the residual is defined to be $y_i - \hat{y}_i$ the distance from the original data point to the predicted value on the line. This can be considered as the error left over after the model has done its work. If the model is doing a good job, then it should explain most of the difference from y, and the first term should be bigger than the second term. If the second term is much bigger, then the model is probably not as useful. By squaring the quantity on the left, and using some algebra and some facts about linear regression, we'll find that where "SS" stands for "sum of squares".

$$\sum_i = \sum_i (\hat{y}_i - \bar{y})^2 + \sum_i (y_i - \hat{y}_i)^2$$

These terms are often abbreviated as SST, SSM, and SSE respectively. If we divide through by SST, we obtain

$$1 = \frac{SSM}{SST} + \frac{SSE}{SST}$$

where we note that r^2 is precisely the coefficient of determination mentioned earlier. One way to evaluate a model's performance is to compare the ratio *SSM/SSE*. It could be done by considering the mean values, *MSM = SSM/(p– 1)*and *MSE = SSE/(n - p),*where the denominators correspond to the degrees of freedom. These new variables MSM and MSE have2 distributions, and their ratio

$$f = \frac{MSM}{MSE}$$

has what's known as an *F* distribution with parameters *(p– 1)*and *(n - p).* The widely used ANOVA test for categorical data. It is based on this F statistic: it's a way of measuring how much of the variability in the data is from the model and how much is from random error, and comparing the two.

The supervised learning pipeline described in previous chapter is implemented using the Python programming interface to Spark (pySpark). PySpark provides

an easy-to-use programming abstraction and parallel runtime. Resilient Distributed Datasets (RDDs) are the key concept. The RDD is the primary abstraction in Spark. It is immutable once constructed. It tracks lineage information to efficiently recomputed lost data and enable operations on collection of elements in parallel. An RDD is created from a data source. The transformations (like map, filter) are applied to an RDD. Finally the actions (like collect, count) are applied to an RDD. The implementation of the system, the experiments conducted, and the results obtained are presented.

B. READ AND PARSE THE INITIAL DATASET

The raw information is presently put away in text document. This crude information is put away in as a RDD, with every component of the RDD addressing an information point as a comma-delimited string. Each string begins with the mark (a year) trailed by mathematical sound highlights. In MLlib, marked preparing examples are put away utilizing the Labeled Point object. The parse Point work is composed that takes as information a crude information point, parses it, and returns a Labeled Point. The crude highlights for 50 information focuses are envisioned in Figure 7.1 by creating a heatmap that imagines each component on a dark scale and shows the variety of each element across the 50 example information focuses. The highlights are all somewhere in the range of 0 and 1, with values more like 1 addressed by means of more obscure shades of grey.

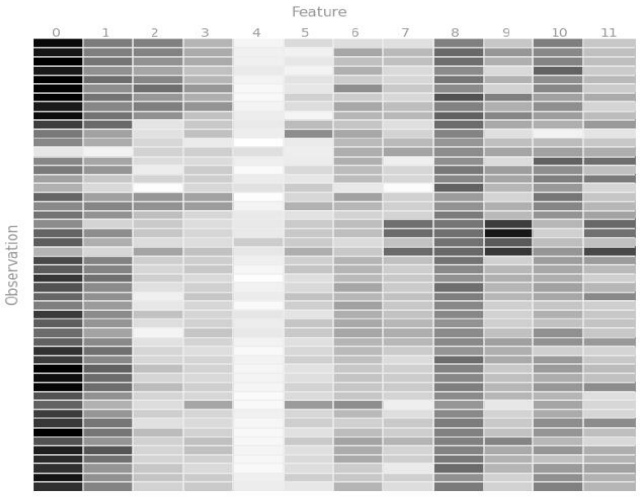

Figure 3.1 Features of raw data

The names inspected to discover the scope of melody years. To do this, first parse every component of the raw data RDD, and afterward track down the littlest and biggest names. The names are a long time during the 1900s and 2000s. In learning issues, it is regularly normal to move marks to such an extent that they start from nothing. Make another RDD comprising of Labelled Point objects in which the marks are moved with the end goal that littlest name rises to nothing.

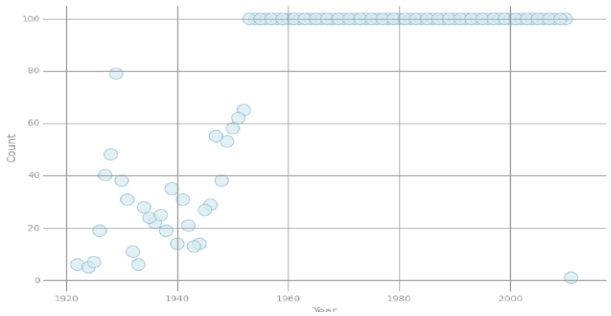

Figure 3.2 Shifting labels

The names when moving are envisioned in Figure 3.2. In the wake of parsing the dataset, and the last undertaking includes split it into preparing, approval, and test sets with the predetermined loads and seed to make RDDs putting away each of these datasets.

C. CREATE AND EVALUATE A BASELINE MODEL

A straightforward yet characteristic benchmark model is one which makes a similar expectation autonomous of the given information point, utilizing the normal mark in the preparation set as the steady forecast esteem. Figure this worth, which is the normal (moved) tune year for the preparation set. To assess the exhibition of this gullible standard model root mean squared mistake (RMSE) is determined for the preparation, approval, and test sets. Figure

3.3 envisions forecasts on the approval dataset.

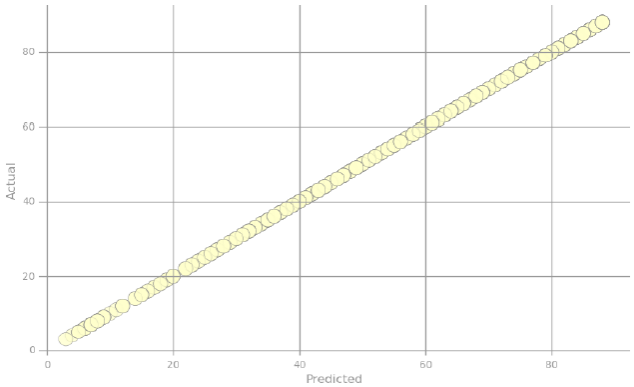

Figure 3.3 Predictions on the validation set

D. **TRAIN AND EVALUATE A LINEAR REGRESSION MODEL**

Train a linear regression model on preparing information and assess its exactness on the approval set. Figure 3.4 shows the log of the preparation mistake as an

element of emphasis. The scatter plot imagines the logarithm of the preparation mistake for every one of the 50 emphasess. The plot shows the preparation mistake itself, zeroing in on the last 44 cycles. Make an expectation on an example point. Next assess the exactness of this model on the approval set.

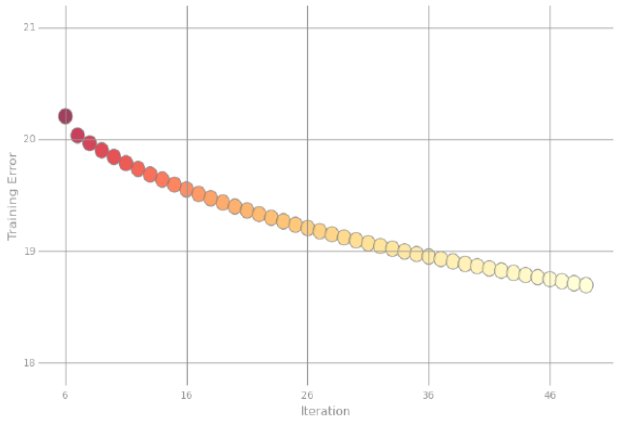

Figure 3.4 Training Error

E. TEST THE FINAL LINEAR REGRESSION MODEL

The last advance is to assess the new model on the test dataset. Since the test set isn't utilized to assess any of the models, the assessment gives a fair gauge for how the built model will perform on new information. Figure 3.5 shows the RMSE for both the pattern model and the new model. This data shows how much preferable the new model performs over the standard model.

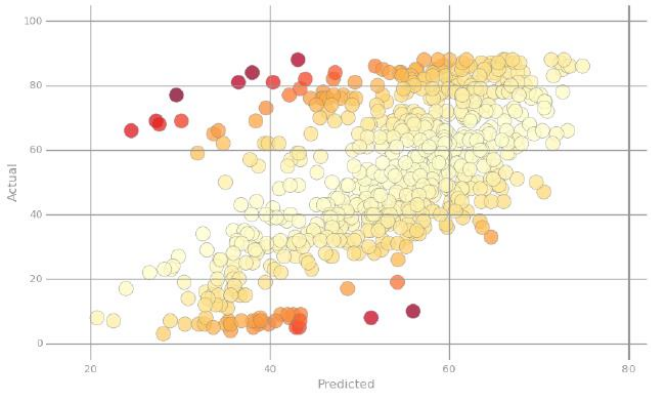

Figure 3.5 Best model's prediction

IV. PERFORMANCE METRICS

Execution of the Root Mean Square Error for both the baseline model and linear regression model is appeared in table 4.1. From this linear regression model performs better compared to the baseline model

Table 4.1 Performance of Root Mean Square Error

Model	RMSE
Baseline	22.137
Linear Regression	16.327

V. CONCLUSION

Supervised learning based framework, MFCC and direct relapse models are proposed to foresee the delivery year of the melody. In this work sound highlights are planned to the delivery year of the tune. At that point this information is parted into preparing set and test set. The preparation set is to prepare the model and the test set is to assess the exactness of the prepared model. Test set assesses last model's exactness and afterward last model can be utilized to mention expectations on future observable facts, new melodies in this work. To assess the presentation of this pattern model Root Mean Squared Error (RMSE) is determined for the preparation, approval, and test sets. The exhibition of RMSE is utilized for both gauge and direct relapse model, yet straight relapse model is better compared to other. Execution of the delivery year of the tune is finished with the assistance of direct relapse model and this model was assessed for a great many melodies, and this strategy accomplishes about generally 90.0% precision. In future work, different component extraction and classifiers strategies will be proposed, to foresee the more assignments in MIR and the presentation will be improved by utilizing different calculation.

REFERENCES

[1] Thierry Bertin-Mahieux, Daniel P.W.Elliis, Brian Whitman and Paul Lamere (2011), "The Million Song Dataset",12th International society for Music Information Retrieval Conference (ISMIR).

[2] Zhouyu Fu, Guojun Lu, Kai Ming Ting and Dengsheng Zhang (2011), "A Survey of Audio-Based Music Classification and Annotation", IEEE Transactions on Multimedia, Vol. 13 No.2.

[3] George Tzanetakis, Georg Essl and Perry Cook (2011), "Automatic Musical Genre Classification of Audio Signals".

[4] Cory McKay CIRMMT and Ichiro Fujinaga CIRMMT (2010), "Improving Automatic Music Classification Performance by Extracting Features from Different Types of Data".

[5] Y.M.D Chathuranga and K.L. Jayaratne (july 2013), "Automatic Music Genre Classification of Audio Signals

with Machine Learning Approches" GSTF International Journal on Computing (joc), vol.3 no.2.

[6] Cory McKay, John Ashley Burgoyne, Jason Hockman(2011), "Evaluating The Genre Classification Performance of Lyrical Features Relative to Audio, Symbolic and Cultural Features

[7] George Tzanetakis, Perry Cook (2002), "Musical Genre Classification of Audio Signals" IEEE Transactions on Speech and Audio Processing, Vol. 10, No.5

[8] Rudolf Mayer and Andreas Rauber (2011), "Musical Genre Classification by Esembles of Audio and Lyrics Features" 12th International Society for Music Information Retrieval Conference (ISMIR).

[9] Francisco Raposo,Richardo Riberio and David Martins (2016), "Using Generic Summarization to improve Music Information Retrieval Tasks" IEEE/ACM Transactions on Audio, Speech and Language Processing, vol.24, No. 6

[10] Loris Nanni, Yandre M.G Costa and Alexander Lumini (2015), "Combining Visual and Acoustic Features for music genre Classifiaction".

[11] Humberto Corona and Michaelmas P.O'Mahony (2015), "An Exploration of Mood Classification in the Million Song Dataset".

[12] Yading song, Simon Dixon, Marcus Pearce (2012), "Evaluation of Musical features for Emotion Classification" 13th International Society for Music Information Retrieval (ISMIR).

[13] Adit Jamdar, Jessica Abraham, Karishma Khanna and Rahul Dubey (May 2015), "Emotion Analysis of Songs Based on Lyrical and Audio Features" International Journal of Artifical Intelligence and Applications (IJAIA) vol.6,no.3.

[14] Cyril Laurier, Olivier Lartillot, Tuomas Eerola and Perti Toiviainen (2012), "Exploring Relationships between Audio Features and Emotion in Music".

[15] S. Palanivel and N.J. Nalini (2013), "Emotion Recognition in Music Signal using AANN and SVM", International Journal of Computer Applications(0975-8887),vol-77 no-2.

[16] Susheel Sharma, Rakesh Singh jadon (2014), "Mood Based Music Classification",International Journal of Innovative Science, Engineering and Technology (IJISET),vol 1,issue 6.

[17] T.N.Charanya, R.Vijayalakshmi (2015),"Music Emption Recognition using Support Vector Machines and Regression Approach",International Journal of Advanced Research in Computer and Communication Engineering vol.4 issue 1.

[18] Katherine Ellis, Emanuele Coviello and Gert R.G. Lanckriet(2011), "Semantic Annotation and Retrieval of Music using a Bag of Systems Representation" 12th International Society for Music Information Retrieval Conference (ISMIR).

[19] Douglas Turnbull, Luke Barrington (2008), "Semantic Annotation and Retrieval of Music and Sound Effects" IEEE Transactions on Audio,Speech and Language Processing, vol 16, No.2.

[20] Thierry Bertin-Mahieux and Daniel P.W.Ellis "Large-Scale Cover Song Recognition using the 2D Fourier Transform Magnitude" 13th International Society for Music Information Retrieval conference (ISMIR).